APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/535,765	03/28/2000	Vincent E. Hummel		5136

7590 06/03/2004	EXAMINER
Blakely Sokoloff Taylor & Zafman LLP	HUISMAN, DAVID J

Blakely Sokoloff Taylor & Zafman LLP
12400 Wilshire Boulevard
7th Floor
Los Angeles, CA 90025

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 06/03/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>26 April 2004</u>.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>22-42</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>22-42</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on <u>26 April 2004</u> is/are: a)☐ accepted or b)☒ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1.   Claims 22-42 have been examined.

Papers Submitted

2.   It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 4/26/2004.

Drawings

3.   The drawings are objected to for the following reason: In the paragraph appearing on page line, beginning on line 19 and continuing to page 10, line 3, it is appears as if each entry within the history table has a corresponding replacement field. However, the amended Fig. 1, shows just a random single box which is supposed to represent the replacement field. The drawing should be amended more appropriately such that each history field has a corresponding replacement field. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Objections

4.   Claim 30 is objected to because of the following informalities: Please insert --a-- before "history multiplexer" in the fourth to last line. Appropriate correction is required.

5.   Claim 37 is objected to because of the following informalities: Please insert --a-- before "history multiplexer" in the fourth to last line. Appropriate correction is required.

Claim Rejections - 35 USC § 112

6.   The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

7.   Claim 31 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention without undue experimentation. According to claim 31, if the first prediction is incorrect, then a replacement field for **said matching entry** is updated, thereby indicating that the entry is used to make a prediction. However, according to independent claim 27, the matching entry is already providing a prediction, and therefore, it has already been indicated that the entry is used to make a prediction. Overall, it is not clear why the updating of the replacement field indicates that the matching entry is used to make a prediction when it is already making a prediction. For purposes of the examination, the matching entry will be interpreted as the entry in the local history table at which the branch instruction information will be written to.

Claim Rejections - 35 USC § 102

8.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9.   Claims 22-23, 27-29 and 40-41 are rejected under 35 U.S.C. 102(e) as being anticipated

by McFarling, U.S. Patent No. 6,374,349 (as applied in the previous Office Action).

10.   Referring to claim 22, McFarling has taught a method comprising:

a) providing a first taken/not-taken prediction responsive to an address using a saturating counter

branch predictor. See column 9, lines 10-16. It has been disclosed that a bimodal predictor

always provides a provisional prediction, the prediction being provisional because it is provided

for use by the system but the system may not choose the prediction as the final prediction.

Furthermore, from column 3, lines 11-28, the bimodal predictor is shown to be an array of

saturated counters.

b) providing (1) a second taken/not-taken prediction responsive to the address resulting in a hit in

a local branch history table, and (2) a hit/miss indication for the address. See column 9, lines 18-

24, and Fig.10, and note that a second prediction is provided (line 102 in Fig.10).

c) the second prediction is generated by a decision function based on a history value from a

matching entry of the local branch history table, the decision function being implemented by

combinational logic that has no memory so as to reduce on-chip area. It should be realized that

the second prediction is generated as the final prediction by the multiplexer shown in Fig.10

when there is a hit in the local table. This multiplexer is combinational logic without a memory

that performs the decision function:

$$pred = (pred_B \text{ AND } hit') \text{ OR } (pred_H \text{ AND } hit)$$

More specifically, the final prediction (pred) is either the prediction from the bimodal predictor (pred$_B$)if there is miss (hit') in the local history table or the prediction from the local history table (pred$_H$) if there is a hit in the local history table. And, combinational logic is just a logic circuit whose outputs depend only on the present logic inputs. Clearly, the multiplexer of Fig.10, outputs a value which is dependent on its inputs (see the formula above).

d) selecting for the address one of (1) the second prediction if the indication is a hit, and (2) the first prediction if the indication is a miss. Again, see column 9, lines 10-24, and Fig.10. If a history-table miss occurs, the bimodal counter prediction will be used. On the other hand, if a history-table hit occurs, the second prediction from the history table will be generated as the final prediction.

11.   Referring to claim 23, McFarling has taught a method as described in claim 22. McFarling has further taught hashing the address prior to indexing at least one of the saturating counter branch predictor and the local branch history table. See column 3, lines 12-16, and column 4, lines 15-19. Recall that hashing is simply the function of converting a value (in this case, the branch instruction address) into a value used for locating corresponding data in a structure (in this case, the saturating counter predictor and the local history predictor). In the system of McFarling, a portion of a full branch instruction address is used to access a prediction entry in a corresponding structure would be considered hashing.

12.   Referring to claim 27, McFarling has taught a processor comprising:

a) an instruction pointer (IP) generator capable of providing an address. See column 3, lines 12-16. Note that McFarling makes reference to a program counter, which performs the same

function as an IP generator (i.e. they both provide an address at which an instruction will be

fetched).

b) saturating counter branch prediction (SCBP) logic having an input coupled to the IP generator

and capable of providing a first taken/not-taken prediction at an output responsive to the address.

See column 9, lines 10-16. It has been disclosed that a bimodal predictor always provides a

prediction. Furthermore, from column 3, lines 11-28, the bimodal predictor is shown to be an

array of saturated counters. Furthermore, column 3, lines 12-16, explain that the SCBP logic is

coupled to the program counter (IP generator).

c) local branch history prediction (LBHP) logic having an input coupled to the IP generator and

capable of providing (1) a second taken/not-taken prediction at an output responsive to the

address resulting in a hit, and (2) a hit/miss indication for the address, wherein the LBHP logic

includes at least one local branch history table to provide a taken/not-taken history from a

matching entry in said table in response to a hit. See column 9, lines 18-24, and Fig.10, and note

that a second prediction is provided (line 102 in Fig.10). Also, in column 4, lines 15-19,

McFarling has explained that the local history table is indexed based on a portion of the branch

instruction address (tag). Therefore, the local history table is also coupled to the IP generator. It

should be realized that the LBHP includes at least one local branch history table to provide a

taken/not-taken history in response to a hit. From Fig.10, it can be seen that the tag/history data

is stored in one table 100 and the prediction (CNT) is stored in a second table. The most

significant bit of CNT is used to make a taken/not-taken prediction. Finally, it should be realized

that the taken/not-taken history is provided from a matching entry (storage location) in the local

table. This matching entry is the location that the branch instruction's address maps to.

d) combinational logic that is predetermined at the time the processor is built to implement

without memory a decision function whose output is the second prediction based on the

taken/not-taken history. It should be realized that the second prediction is generated as the final

prediction by the multiplexer shown in Fig. 10 when there is a hit in the local table. This

multiplexer is combinational logic without a memory that performs the decision function:

$$pred = (pred_B \text{ AND } hit') \text{ OR } (pred_H \text{ AND } hit)$$

More specifically, the final prediction (pred) generated by the multiplexer is either the prediction

from the bimodal predictor ($pred_B$) if there is miss (hit') in the local history table or the

prediction from the local history table ($pred_H$) if there is a hit in the local history table. And,

combinational logic is just a logic circuit whose outputs depend only on the present logic inputs.

Clearly, the multiplexer of Fig. 10, outputs a value which is dependent on its inputs (see the

formula above).

e) a multiplexer having an input coupled to the outputs of the SCBP and LBHP logic and a select

input coupled to receive the hit/miss indication and in response provide (1) the second prediction

if there is a hit and (2) the first prediction if there is a miss. See Fig. 10 and note that the

multiplexer has inputs from both the SCBP (104) and LBHP (104) and that one of the inputs is

chosen based on the hit signal that is used as a select line for the multiplexer. More specifically,

from column 9, lines 10-24, if a history-table miss occurs, the provisional bimodal counter

prediction will be used. On the other hand, if a history-table hit occurs, the second prediction

from the history table will be generated as the final prediction.

13.   Referring to claim 28, McFarling has taught a processor as described in claim 27.

McFarling has further taught address hash logic coupled between the IP generator and the inputs

of the SCBP and LBHP logic to provide a plurality of index values to at least one of the SCBP

and LBHP logic. Again, as discussed in column 3, lines 12-16, and column 4, lines 15-19, the

prediction logic is provided an index by performing a truncation hashing function. More

specifically, it has been taught (in the above passages) that the prediction logic receives a certain

amount of low order branch instruction address bits as an index instead of the entire address.

Therefore, the subset of wires used to carry the index from the IP generator to the prediction

logic is address hash logic that performs a truncation algorithm. This logic transforms an initial

value (branch instruction address) into another value used for locating corresponding data in a

structure (SCBP and LBHP).

14.   Referring to claim 29, McFarling has taught a processor as described in claim 27.

McFarling has further taught that the SCBP logic includes a bimodal predictor. See column 9,

lines 10-24.

15.   Referring to claim 40, McFarling has taught a method comprising:

a) providing a first taken/not-taken prediction responsive to an address using a saturating counter

branch predictor. See column 9, lines 10-16. It has been disclosed that a bimodal predictor

always provides a prediction. Furthermore, from column 3, lines 11-28, the bimodal predictor is

shown to be an array of saturated counters.

b) providing a second taken/not-taken prediction responsive to the address resulting in a hit in a

local branch history table. See column 9, lines 18-24, and Fig.10. Note in Fig.10 that if a hit in

the local table occurs, a second prediction is provided on line 102.

c) the second prediction is generated using a decision function based on a taken/not-taken history value from a history field of a matching entry of the local branch history table, the decision function being implemented by combinational logic that has no counter or state machine. It should be realized that the second prediction is generated as the final prediction by the multiplexer shown in Fig.10 when there is a hit in the local table. This multiplexer is combinational logic without a counter or state machine that performs the decision function:

$$pred = (pred_B \text{ AND hit'}) \text{ OR } (pred_H \text{ AND hit})$$

More specifically, the final prediction (pred) generated by the multiplexer is either the provisional first prediction from the bimodal predictor ($pred_B$) if there is miss (hit') in the local history table or the second prediction from the local history table ($pred_H$) if there is a hit in the local history table. And, combinational logic is just a logic circuit whose outputs depend only on the present logic inputs. Clearly, the multiplexer of Fig.10, outputs a value which is dependent on its inputs (see the formula above).

d) selecting for the address one of the first prediction and the second prediction depending on whether or not the address resulted in said hit. See column 9, lines 10-24, and Fig.10. If a history-table miss occurs, the counter prediction will be used. On the other hand, if a history-table hit occurs, the history table prediction will be used.

16. Referring to claim 41, McFarling has taught a method as described in claim 40. McFarling has further taught:

a) updating a replacement field for an existing entry in the local branch history table that includes a history field containing a plurality of taken/not-taken outcomes of one or more previously

executed branch instructions associated with said address, only if the first prediction is incorrect.

It should be noted that if there is a miss in the local table, then the bimodal predictor (saturating

counter predictor) will make the prediction. According to column 9, lines 45-50, if the bimodal

prediction is already correct, then no replacement will occur to the local table. Therefore,

replacements will only occur when the first prediction from the bimodal predictor is incorrect.

When it is incorrect, an entry within the local table (a replacement field which includes a tag,

history, and valid field as shown in Fig. 17) will be replaced with information corresponding to

the mispredicted branch. In other words, each entry within the table is a replacement field.

Whatever entry is replaced is the matching entry as the information for the new branch will only

be placed into a single entry. This entry would be determined by some replacement algorithm

such as the least recently used (LRU) replacement algorithm. This is mentioned in column 8,

lines 6-8.

Claim Rejections - 35 USC § 103

17.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

18.   Claims 25-26 and 32-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over

McFarling, as applied above, in view of Hennessy and Patterson, Computer Architecture - A

Quantitative Approach, 2nd Edition (as applied in the previous Office Action and herein referred

to as Hennessy).

19.   Referring to claim 25, McFarling has taught a method as described in claim 22.

a) Furthermore, it is inherent that an instruction for which a prediction is made would have been

fetched at the corresponding address. This concept is also supported in column 3, lines 12-16.

The bits that are used to index the predictors are taken from an address that is stored in the

program counter (PC). The PC is a register that is used to store a pointer to a location in memory

from which an instruction is fetched.

b) In addition, McFarling has taught that if the instruction is a branch, a determination as to

whether the branch is taken or not-taken will not be available until the instruction has progressed

beyond a decode stage. See column 1, lines 33-36, and note that the outcome of the branch is

unknown until after the branch is executed. Since execution inherently follows decoding, the

branch determination is not known until the instruction has progressed beyond the decode stage.

c) Finally, although it is inherent that McFarling's system decodes fetched instructions (in order

to determine what operation would be performed), McFarling has not explicitly taught that at

least one of the first and second predictions is available when the at least one instruction is being

decoded. However, Hennessy has shown the state of a pipeline that implements branch

prediction. See Figure 3.26 on page 167. Note that during the decode stage (ID) of the branch

instruction, the predicted instruction (Instruction i+1) is being fetched in the fetch stage (IF). In

order for the predicted instruction to be fetched during the decode stage of the branch instruction,

the prediction must be available when the branch is being decoded. It can be seen that the

pipeline is kept full and free of stalls through the use of branch prediction. Furthermore, from

column 1, lines 49-56, McFarling has explained that the purpose of a branch prediction scheme

is to keep the pipeline full and free from stalls, which in fact is supported by the teachings of

Hennessy. Therefore, it would have been obvious to one of ordinary skill in the art at the time of

the invention to make the at least one of the predictions available when the at least one

instruction is being decoded. This will ensure that the pipeline will stay full, as is desired by

McFarling. It should be further noted that the scheme taught by Hennessy does not only apply to

a system in which branch outcomes are known at the end of the decode stage. This can be

realized since McFarling has taught a taken/not-taken prediction scheme in which branch

outcomes are unknown until after the branch executes (column 1, lines 33-45). The general idea

of such a prediction scheme is to keep the pipeline full regardless of when the branch outcome is

"officially" known, and if the prediction is available while the branch is being decoded, then the

next instruction can be fetched without stalling.

20.   Referring to claim 26, McFarling in view of Hennessy has taught a method as described

in claim 25. If the at least one instruction is a branch, then:

a) it is inherent that the method further comprises determining a target address of the branch. If a

target address were not determined for a conditional branch, then the system would not know

where to fetch the next instruction from when the branch is predicted taken.

b) it is inherent that the method further comprises loading an instruction pointer generator with

the target address if at least one of the first and second predictions indicates that the branch is to

be taken. In order to fetch the target instruction of a predicted-taken branch, the target address

must be loaded into the IP generator (also known as the program counter (PC)). The IP

generator (PC) is a register that is used to store a pointer to a location in memory from which an

instruction is fetched. For the applicant's benefit, please see Hennessy, page 162 and Figure

3.22 on page 163. From the figure it can be seen that the PC (IP generator) is loaded with the

output of a multiplexer. Inputs to the multiplexer include the next instruction address and the

target address (which is outputted by the ADD logic in the ID stage of the pipeline). If the target

address were not stored in the IP generator (PC), then the target instruction could not be fetched

from instruction memory (as shown in Figure 3.22 of Hennessy).

21.   Referring to claim 32, McFarling has taught a processor as described in claim 27.

Furthermore, the processor of claim 32 performs the method described in claim 25. Therefore,

claim 32 is rejected for the same reasons set forth in the rejection of claim 25.

22.   Referring to claim 33, McFarling has taught a processor as described in claim 32.

Furthermore, the processor of claim 33 performs the method described in claim 26. In addition,

it should be realized from the rejection of claims 25 and 26 that the decode stage taught by

Hennessy has the ability to determine a target address. See Figure 3.22 on page 163. Therefore,

claim 33 is rejected for the same reasons set forth in the rejection of claim 26.

23.   Claim 30 is rejected under 35 U.S.C. 103(a) as being unpatentable over McFarling, as

applied above, in view of Gochman et al., U.S. Patent No. 5,842,008 (as applied in the previous

office action and herein referred to as Gochman).

24.   Referring to claim 30, McFarling has taught a processor as described in claim 27.

a)   1) McFarling has taught a single branch history table that provides a tag and a taken/not-

taken history associated with the tag in response to a hit. See Fig.10. McFarling has not

explicitly taught the concept of using multiple branch history tables to provide a tag and a

taken/not taken history associated with the tag in response to a hit. However, Gochman has

taught the concept of using multiple memories that are simultaneously accessed based on an

index. After each memory produces its respective data, a single value is selected and

propagated. See Fig.4a and Fig.5. A person of ordinary skill in the art would have recognized

that by using multiple memory tables, as opposed to a single table, the number of index bits used

to access an entry in each bank would be reduced, resulting in a reduction in the amount of

hardware. Therefore, it would have been obvious to one of ordinary skill in the art at the time of

the invention to use multiple branch history tables instead of a single table.

2) In addition, it would have been obvious to one of ordinary skill in the art at the time of

the invention to separate the single history table as taught by McFarling into multiple history

tables since it has been held that making separable where needed is obvious. See <u>Nerwin v.</u>

<u>Erlichman</u>, 168 USPQ 177 (1969). It should be realized that the single-table system serves the

same purpose as the multiple-table system in that an index is applied and a corresponding entry

is selected. Although multiple entries are selected in response to applying an index to multiple

tables (in Applicant's system), the final result is still only a single entry, as taught by McFarling.

b) it is inherent that McFarling's system includes compare logic that is connected to the history

table in order to determine the hit/miss indication. A hit indication is given when the branch

instruction in question has a corresponding entry in the history table. Therefore, in order to give

a hit signal, some logic must be used to compare the entries in the table with the current branch

instruction. Since McFarling has not taught multiple history tables, it follows that he has not

taught multiple compare logic units that are connected to those tables. However, if multiple

tables were used, as discussed in part (a) above, then it would be inherent that multiple compare

logic units would be connected to those tables in order to determine if a branch has a

corresponding entry.

c) McFarling has not taught a history multiplexer coupled to each of the plurality of tables to

provide the history for the hit. However, since McFarling has only taught one table, there is no

need to multiplex multiple values from multiple tables. On the other hand, if multiple tables

were implemented in McFarling's system, as described in part (a) above, then it would follow

that a multiplex system would be needed in order to choose one of the plurality of history values.

This concept has been taught by Gochman in Fig.4a (component 320) and Fig.5. When each of

the values are retrieved from the memories, the selection logic determines which value will be

sent through. Therefore, if multiple tables were used in McFarling's system, in order to ensure

only the correct history value is used to make the prediction, it would have been obvious to one

of ordinary skill in the art at the time of the invention to use a multiplexer to perform the

selection.

d) McFarling has taught combinational logic to provide the second taken/not-taken prediction.

See Fig.10, and note the multiplexer which selects between either the local prediction or the

bimodal prediction. However, since McFarling has not taught multiple history tables and a

history multiplexer, it follows that he has not taught combinational logic that is connected to a

history multiplexer to provide the second taken/not-taken prediction. It should be realized

though that the combinational logic of McFarling takes in the second prediction as an input. If

multiple tables and a history multiplexer were used, as discussed in parts (a) and (c) above, then

it would follow that the combinational logic would be coupled to an output of the history

multiplexer because the history multiplexer will be providing the second taken/not-taken

prediction as an input to the combinational logic. Therefore, if multiple tables and a history

multiplexer were used, it would have been obvious to one of ordinary skill in the art at the time

of the invention to connect the combinational logic taught by McFarling to the output of the

history multiplexer since the combinational logic uses the history information in order to

generate a taken/not-taken prediction.

25.   Claim 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over McFarling, as

applied above, in view of Rahman et al., U.S. Patent No. 5,805,878 (as applied in the previous

Office Action and herein referred to as Rahman).

26.   Referring to claim 34, McFarling has taught a processor as described in claim 27.

McFarling has not explicitly taught that the address points to a cache line having a plurality of

instructions. However, Rahman has taught the implementation of a cache with multiple

instructions per cache line. See column 9, lines 17-21. Also, Rahman has suggested in column

9, lines 23-37, that a benefit of such a feature would be to allow for simultaneous fetching of

multiple instructions. This would result in less time spent making memory accesses and

increased instruction-level parallelism, which would result in higher throughput. Therefore, in

order to increase throughput, it would have been obvious to one of ordinary skill in the art at the

time of the invention to implement a cache that contains a plurality of instructions per cache line.

27.   Claims 24, 31, 35-36, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable

over McFarling, as applied above, in view of Henry et al., U.S. Patent No. 6,550,004 (as applied

in the previous Office Action and herein referred to as Henry).

28.   Referring to claim 24, McFarling has taught a method as described in claim 22.

McFarling has further taught:

a) updating a replacement field for a matching entry in the local branch history table only if the first prediction is incorrect, indicating that the entry is used to make a prediction. It should be noted that if there is a miss in the local table, then the bimodal predictor (saturating counter predictor) will make the prediction. According to column 9, lines 45-50, if the bimodal prediction is already correct, then no replacement will occur to the local table. Therefore, replacements will only occur when the first prediction from the bimodal predictor is incorrect. When it is incorrect, an entry within the local table (a replacement field which includes a tag, history, and valid field as shown in Fig. 17) will be replaced with information corresponding to the mispredicted branch. Whatever entry is replaced is the matching entry as the information for the new branch will only be placed into a single entry. This entry would be determined by some replacement algorithm such as the least recently used (LRU) replacement algorithm. This is mentioned in column 8, lines 6-8. This update indicates that the matching entry will be used to make a prediction in the future because the next time that same branch is encountered, a hit will occur in the local history table and the local table will provide the prediction as opposed to the incorrect bimodal prediction.

b) McFarling has not explicitly taught means for updating said history field, with the outcome of an executed branch instruction, only if the first prediction is incorrect and the second prediction is correct. However, Henry has taught such a concept. See Fig. 6 and note that if the final (first) prediction is incorrect (step 624) and the non-selected (second) prediction is correct (step 628), then the corresponding history field within predictor 212 (Fig. 2) will be updated (step 632). It should be noted from Fig. 3 that component 212 does have history entries. Henry has further disclosed that such an update scheme advantageously yields improved branch prediction results.

As a result, it would have been obvious to one of ordinary skill in the art at the time of the

invention to modify McFarling to employ the update policy taught by Henry.

29.   Referring to claim 31, McFarling has taught a method as described in claim 27.

McFarling has further taught:

a) logic to update a replacement field for said matching entry only if the first prediction is

incorrect, indicating that the entry is used to make a prediction. It should be noted that if there is

a miss in the local table, then the bimodal predictor (saturating counter predictor) will make the

prediction. According to column 9, lines 45-50, if the bimodal prediction is already correct, then

no replacement will occur to the local table. Therefore, replacements will only occur when the

first prediction from the bimodal predictor is incorrect. When it is incorrect, an entry within the

local table (a replacement field which includes a tag, history, and valid field as shown in Fig.17)

will be replaced with information corresponding to the mispredicted branch. Whatever entry is

replaced is the matching entry as the information for the new branch will only be placed into a

single entry. This entry would be determined by some replacement algorithm such as the least

recently used (LRU) replacement algorithm. This is mentioned in column 8, lines 6-8. This

update indicates that the matching entry will be used to make a prediction in the future because

the next time that same branch is encountered, a hit will occur in the local history table and the

local table will provide the prediction as opposed to the incorrect bimodal prediction.

b) McFarling has not explicitly taught logic to update a history field for said matching entry with

the outcome of an executed branch instruction only if the first prediction is incorrect and the

second prediction is correct. However, Henry has taught such a concept. See Fig.6 and note that

if the final (first) prediction is incorrect (step 624) and the non-selected (second) prediction is

correct (step 628), then the corresponding history field within predictor 212 (Fig.2) will be

updated (step 632). It should be noted from Fig.3 that component 212 does have history entries.

Henry has further disclosed that such an update scheme advantageously yields improved branch

prediction results. As a result, it would have been obvious to one of ordinary skill in the art at

the time of the invention to modify McFarling to employ the update policy taught by Henry.

30.   Referring to claim 35, McFarling has taught an apparatus comprising:

a) means for providing an address of at least one instruction. See column 3, lines 12-16. Note

that McFarling makes reference to a program counter, which performs the same function as an IP

generator (i.e. they both provide an address at which an instruction will be fetched).

b) means for providing a first taken/not-taken branch prediction based upon the current state of a

state machine and responsive to the address. See column 9, lines 10-16. It has been disclosed

that a bimodal predictor always provides a provisional prediction, because it is provided for use

by the system but the system may not choose the prediction as the final prediction. Furthermore,

from column 3, lines 11-28, the bimodal predictor is shown to be an array of saturated counters.

Saturated counters are state machines in that they function differently when in different states.

See Fig.2.

c) local branch history prediction (LBHP) logic having an input coupled to the address providing

means and capable of providing (1) a second taken/not-taken prediction at an output responsive

to the address resulting in a hit, and (2) a hit/miss indication for the address, wherein the LBHP

logic includes at least one local branch history table to provide a taken/not-taken history in

response to a hit. See column 9, lines 18-24, and Fig.10. Also, in column 4, lines 15-19,

McFarling has explained that the local history table is indexed based on a portion of the branch

instruction address. Therefore, the local history table is also coupled to the IP generator. It

should be realized that the LBHP includes at least one local branch history table to provide a

taken/not-taken history in response to a hit. From Fig.10, it can be seen that the tag/history data

is stored in one table 100 and the prediction (CNT) is stored in a second table. When a hit in the

local table occurs, the most significant bit of accessed CNT is used to make a second taken/not-

taken prediction.

d) means for generating without memory the second prediction based on the taken/not-taken

history. It should be realized that the second prediction is generated as the final prediction by the

multiplexer shown in Fig.10 when there is a hit in the local table. This multiplexer is

combinational logic without a memory that performs the decision function:

$$\text{pred} = (\text{pred}_B \text{ AND hit'}) \text{ OR } (\text{pred}_H \text{ AND hit})$$

More specifically, the final prediction (pred) generated by the multiplexer is either the

provisional prediction from the bimodal predictor (pred_B) if there is miss (hit') in the local

history table or the second prediction from the local history table (pred_H) if there is a hit in the

local history table. And, combinational logic is just a logic circuit whose outputs depend only on

the present logic inputs. Clearly, the multiplexer of Fig.10, outputs a value which is dependent

on its inputs (see the formula above).

e) a multiplexer having an input coupled to the outputs of the first prediction means and the

LBHP logic and a select input coupled to receive the hit/miss indication and in response provide

(1) the second prediction if there is a hit and (2) the first prediction if there is a miss. See Fig.10

and note that the multiplexer has inputs from both the SCBP (104) and LBHP (104) and that one

of the inputs is chosen based on the hit signal that is used as a select line for the multiplexer. See column 9, lines 10-24 for further explanation.

f) McFarling has not explicitly taught means for updating said history field, with the outcome of an executed branch instruction that is pointed to by said address, only if the first prediction is incorrect and the second prediction is correct. However, Henry has taught such a concept. See Fig.6 and note that if the final (first) prediction is incorrect (step 624) and the non-selected (second) prediction is correct (step 628), then the corresponding history field within predictor 212 (Fig.2) will be updated (step 632). It should be noted from Fig.3 that component 212 does have history entries. Henry has further disclosed that such an update scheme advantageously yields improved branch prediction results. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify McFarling to employ the update policy taught by Henry.

31.   Referring to claim 36, McFarling in view of Henry has taught an apparatus as described in claim 35. Furthermore, claim 36 is rejected for the same reasons set forth in the rejection of claim 23. Note that hashing and encoding perform the same operation in that a starting value (branch instruction address) is transformed into another value (index to prediction logic).

32.   Referring to claim 42, McFarling has taught a method as described in claim 41. McFarling has not explicitly taught updating said history field based on an outcome of an executed branch instruction that is pointed to by said address, only if the first prediction is incorrect and the second prediction is correct. However, Henry has taught such a concept. See Fig.6 and note that if the final (first) prediction is incorrect (step 624) and the non-selected (second) prediction is correct (step 628), then the corresponding history field within predictor

212 (Fig.2) will be updated (step 632). It should be noted from Fig.3 that component 212 does

have history entries. Henry has further disclosed that such an update scheme advantageously

yields improved branch prediction results. As a result, it would have been obvious to one of

ordinary skill in the art at the time of the invention to modify McFarling to employ the update

policy taught by Henry.

33.   Claim 37 is rejected under 35 U.S.C. 103(a) as being unpatentable over McFarling in

view of Henry, as applied above, in view of Gochman, as applied above.

34.   Referring to claim 37, McFarling has taught an apparatus as described in claim 35.

Furthermore, claim 37 is rejected for the same reasons set forth in the rejection of claim 30.

35.   Claims 38-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over McFarling

in view of Henry, as applied above, in view of Hennessy, as applied above.

36.   Referring to claim 38, McFarling has taught an apparatus as described in claim 35.

Furthermore, the apparatus of claim 38 performs the method of claim 25. Therefore, claim 38 is

rejected for the same reasons set forth in the rejection of claim 25.

37.   Referring to claim 39, McFarling has taught an apparatus as described in claim 38.

Furthermore, the apparatus of claim 39 performs the method of claim 26. Therefore, claim 39 is

rejected for the same reasons set forth in the rejection of claim 26.

Response to Arguments

38.   Applicant's arguments filed on April 26, 2004, have been fully considered but they are

not persuasive.

39.   Applicant argues that the independent claims are novel because of the newly added

amendments. Although these arguments have been considered, they are not found persuasive for

the following reasons: In summation, looking at Fig. 10 of McFarling, a first prediction is

provided by the bimodal predictor (saturating counter predictor) on line 104. This first

prediction is a provisional prediction because although it is provided for use by the system, the

system may not select this prediction. Similarly, a second prediction is provided by the local

predictor on line 102. In order to decide which prediction to select (102 and 104), a decisional

function is implemented by combinational logic which has no memory, no counter, and no state

machine. More specifically, the multiplexer shown in Fig. 10 is this combinational logic without

a memory, counter, and state machine, that performs the following decision function:

$$pred = (pred_B \text{ AND } hit') \text{ OR } (pred_H \text{ AND } hit)$$

From this formula, the final prediction (pred) is either the first prediction from the

bimodal predictor ($pred_B$) if there is miss (hit') in the local history table, or the second prediction

from the local history table ($pred_H$) if there is a hit in the local history table. And, combinational

logic is just a logic circuit whose outputs depend only on the present logic inputs. Clearly, the

multiplexer of Fig. 10, outputs a value which is dependent on its inputs (see the formula above).

It should be realized that if there is a hit in the history table, then the second prediction is

generated as a final prediction by the decision function. That is, $pred = pred_H$.

Conclusion

40.   Applicant's amendment necessitated the new ground(s) of rejection presented in this

Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a).

Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the date of this

final action.

The prior art made of record and not relied upon is considered pertinent to applicant's

disclosure. Applicant is reminded that in amending in response to a rejection of claims, the

patentable novelty must be clearly shown in view of the state of the art disclosed by the

references cited and the objections made. Applicant must also show how the amendments avoid

such references and objections. See 37 CFR § 1.111(c).

Hunt, U.S. patent No. 6,151,672, has taught methods and apparatus for reducing

interference in a branch history table of a microprocessor. In addition, Fig.2, component 216,

and column 7, lines 11-39 show that additional combinational logic is used to generate a

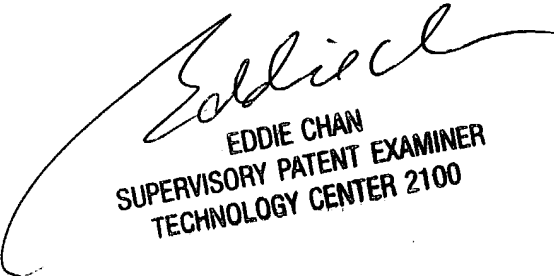prediction based on the history for that branch and a hint for that branch.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
May 28, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100